

The Weight in Enumeration

Johannes Schmidt

Jönköping International Business School, Jönköping University, Sweden
 johannes.schmidt@ju.se

Abstract. In our setting enumeration amounts to generate all solutions of a problem instance without duplicates. We address the problem of enumerating the models of B -formulae. A B -formula is a propositional formula whose connectives are taken from a fixed set B of Boolean connectives. Without imposing any specific order to output the solutions, this task is solved. We completely classify the complexity of this enumeration task for all possible sets of connectives B imposing the orders of (1) non-decreasing weight, (2) non-increasing weight; the weight of a model being the number of variables assigned to 1. We consider also the weighted variants where a non-negative integer weight is assigned to each variable and show that this add-on leads to more sophisticated enumeration algorithms and even renders previously tractable cases intractable, contrarily to the constraint setting. As a by-product we obtain also complexity classifications for the optimization problems known as MIN-ONES and MAX-ONES which are in the B -formula setting two different tasks.

Keywords: Computational Complexity, Enumeration, non-decreasing weight, Polynomial delay, Post's Lattice, MaxOnes

1 Introduction

We deal in this paper with algorithmic and complexity of *enumeration*, the task of generating all solutions of a problem instance. Over the last 15 years, in both practice and theory, one can observe a growing interest in studying enumeration problems which have previously been poorly studied compared to decision, optimization and counting problems. The main reason for this may lie in the huge increase of the size of the data computers are nowadays demanded and able to process in everyday applications.

It is in the meanwhile commonly agreed to consider an enumeration algorithm *efficient* if it has *polynomial delay* ([9,16]), *i.e.*, the time passing between outputs of two successive solutions is polynomial in the input size (while the total time of the output process is usually exponential, due to large solution sets). Variants and different degrees of efficiency in this context exist, see e.g. [9,19]. Known reductions for enumeration are essentially one-to-one parsimonious reductions, as opposed to counting complexity where a greater variety of useful reductions exist, see e.g. [7,3]. An interesting issue of enumeration is the order in which the solutions are output. Imposing different orders for an enumeration process may drastically change the complexity, see e.g. [9,5,2].

We focus in this paper on the task of enumerating the models of a propositional formula. This task has already been addressed in the context of Boolean constraint satisfaction problems (CSPs). One considers here formulæ in generalized conjunctive normal form [15], also called Γ -*formulæ* where Γ is the *constraint language*. In [4] this task, ENUMSAT for short, has been studied without imposing any special order. There is a polynomial delay algorithm if and only if the underlying constraint language Γ is either Horn, or dual Horn, or affine, or 2CNF, unless $P = NP$. It is worth mentioning that the algorithms underlying this result are all straight forward extensions of the corresponding decision procedures via the notion of *self-reducibility* [18] which naturally leads to lexicographic order. In the non-Boolean domain the self-reducible fragment does not deliver all tractable cases anymore [17] and things get much more involved.

Back to the Boolean domain, ENUMSAT has also been considered imposing the order of non-decreasing weight (ENUMSAT_↑ for short), the weight of a model being the number of variables assigned to 1. The weight is a natural parameter in Boolean CSPs that can be assimilated to the cost of an assignment. Hence, the task ENUMSAT_↑ can be seen as the task of enumerating the cheapest solutions first, then the more expensive ones in order of increasing cost. In [5] the task ENUMSAT_↑ has been studied for Γ -formulæ. There is a polynomial delay algorithm to enumerate the models of a propositional Γ -formula by order of non-decreasing weight if and only if Γ is width-2-affine or Horn, unless $P = NP$. By duality in that context, the task of enumerating by order of non-increasing weight, ENUMSAT_↓ for short, is tractable if and only if Γ is width-2-affine or dual Horn.

In this paper we reveal new tractable fragments of propositional logic for ENUMSAT_↑ and ENUMSAT_↓ by considering fragments of propositional logic by a different approach. A *B-formula* is a propositional formula whose connectives are taken from B , a fixed set of Boolean functions. This approach covers different fragments than the classical constraint approach, e.g. monotonic, self-dual, 0-separating of degree n . It has first been taken by Lewis [12] who showed that the satisfiability problem for *B-formulæ*, SAT(B) for short, is NP-complete if and only if the set B is able to express negation of implication ($x \wedge \neg y$), unless $P = NP$. Since then, a number of problems dealing with propositional formulæ have been parameterized by *B-formulæ* in order to get a finer classification of their complexity, e.g. equivalence [14], implication [1], circumscription [20], abduction [6].

In [2] the model enumeration problem has been studied in the context of *B*-circuits without imposing an order and imposing lexicographic order. Roughly speaking, a *B-formula* can be represented by a *B-circuit* without size-increase, but, in general, not vice versa. Therefore, tractability translates from *B*-circuits to *B-formulæ*, whereas this does not automatically hold for hardness results. We observe however that only slight modifications in the hardness proof from [2] suffice and one obtains the same classification for *B-formulæ*.

Our main contribution lies in complete classifications for ENUMSAT_↑ and ENUMSAT_↓. We show that the models of a *B-formula* can efficiently be enumer-

ated by order of non-decreasing weight if and only if the connectives are either 0-separating, or affine, or conjunctive, or disjunctive, unless $P = NP$. We further show that we can efficiently enumerate by order of non-increasing weight if and only if the connectives are either 0-separating of degree 2, or monotone, or affine, unless $P = NP$. We also consider the weighted variants of $\text{ENUMSAT}_{\uparrow}$ and $\text{ENUMSAT}_{\downarrow}$ (denoted $\text{W-ENUMSAT}_{\uparrow}$ and $\text{W-ENUMSAT}_{\downarrow}$, respectively) where a weight function $w : \{x_1, \dots, x_n\} \rightarrow \mathbb{N}$ assigns a non-negative integer weight to each variable and the weight of an assignment is the sum of the weights of the variables assigned to 1. We show that for $\text{W-ENUMSAT}_{\uparrow}$ the previously tractable fragment of 0-separating connectives now compounds intractable cases.

We also shed new light on the optimization problems known as MIN-ONES and MAX-ONES where the task is to find a model of minimal / maximal weight. We use these tasks, together with their weighted and non-trivial variants, to obtain hardness of the enumeration problems. These two tasks are in our setting not "the same": contrary to the classical constraint setting [10], no duality notion allows to easily derive the classification for MAX-ONES from the one for MIN-ONES , or vice versa. This is because the duality notion in our setting transforms MIN-ONES (find a satisfying assignment with minimal number of 1's) into the task of finding a non-satisfying assignment with maximal number of 1's. We show further that allowing weights on the variables renders previously tractable fragments intractable, contrarily to the classical constraint approach.

Among the algorithmic enumeration strategies we use, we apply a method we shall call *priority queue method*. It has first been used in [9] in order to enumerate all maximal independent sets of a graph in lexicographical order. This method turned out to be applicable in much more generality [11,16,5]. We use it to obtain various polynomial delay algorithms for $\text{ENUMSAT}_{\uparrow}$ and $\text{ENUMSAT}_{\downarrow}$ and their weighted variants.

We give another non-trivial enumeration algorithm for $\text{ENUMSAT}_{\downarrow}$ for the fragment of connectives that are 0-separating of degree 2 (Proposition 16) that may be intuitively best described by *nested* or *incremental brute force*: we use the Erdős-Ko-Rado Theorem [8] to obtain a combinatorial bound that allows us to *buy time* [17] from a relatively large number of models whose output process delivers then enough time to compute further, computationally more involving models that are stored and output afterwards.

The paper is organized as follows. In Section 2 we give the necessary preliminaries on complexity theory, propositional formulæ and clones of Boolean functions. In Section 3 we briefly look at model enumeration without order prescription. We treat the order of non-decreasing and non-increasing weight in Sections 4 and 5 respectively. We conclude in Section 6.

2 Preliminaries

2.1 Complexity Theory

For the decision problems the arising complexity degrees encompass the classes P and NP . For our hardness results we employ logspace many-one reductions.

An *enumeration problem* E can be formalized by a triple (I, Sol, \leq) , where I are the instances, Sol is a function mapping each instance $x \in I$ to its set of solutions $Sol(x)$ and \leq is a partial order (possibly empty) on the solution space. We say that an algorithm A solves an enumeration problem $E = (I, Sol, \leq)$ if for a given input $x \in I$, A generates one by one the elements of $Sol(x)$ without repetition such that for all $y, z \in Sol(x)$ such that $y < z$, A outputs y before z .

An enumeration algorithm runs in *polynomial delay* if the delay until the first solution is output and thereafter the delay between any two consecutive solutions is bounded by a polynomial $p(n)$ in the input size n . We denote *DelayP* the class of enumeration problems that admit a polynomial delay algorithm and *SpaceDelayP* those problems in DelayP that are solvable within polynomial space.

2.2 Propositional Formulæ

We assume familiarity with propositional logic. For a propositional formula φ we denote by $\text{Vars}(\varphi)$ the set of variables occurring in φ . We represent an assignment $\sigma : \text{Vars}(\varphi) \rightarrow \{0, 1\}^n$ usually as a tuple over $\{0, 1\}$ or when convenient by the set of variables assigned to 1, *i.e.*, the empty set corresponds to $\mathbf{0}$ and $\text{Vars}(\varphi)$ to $\mathbf{1}$. A *model* for a formula φ is an assignment that satisfies φ . A *non-trivial* assignment is an assignment different from $\mathbf{0}$ and $\mathbf{1}$. The *complement* of an assignment σ is defined as $\bar{\sigma}(x) = 0 \Leftrightarrow \sigma(x) = 1$. We call a variable $x \in \text{Vars}(\varphi)$ *fictive*, if the assignment $x = 0$ can be extended to a model of φ if and only if so can the assignment $x = 1$. We denote by $\varphi[\alpha/\beta]$ the formula obtained from φ by replacing all occurrences of α with β .

2.3 Clones of Boolean Functions

A *Boolean function* is an n -ary function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. For technical reasons we consider only Boolean functions of arity > 0 . It is not difficult but just technical to include also functions of arity 0 into our considerations. We denote the n -ary Boolean constants by C_0^n and C_1^n , respectively. When the arity is not relevant, we indicate them also by C_0 and C_1 , keeping in mind that they have at least one fictive coordinate. An n -ary assignment m such that $f(m) = 1$ will be called *model* of f . A *clone* is a set of Boolean functions that is closed under superposition, *i.e.*, it contains all projections (that is, the functions $f(a_1, \dots, a_n) = a_k$ for all $1 \leq k \leq n$ and $n \in \mathbb{N}$) and is closed under arbitrary composition. Let B be a finite set of Boolean functions. We denote by $[B]$ the smallest clone containing B and call B a *base* for $[B]$. In 1941 Post identified the set of all clones of Boolean functions [13]. He gave a finite base for each of the clones and showed that they form a lattice under the usual \subseteq -relation, hence the name *Post's lattice* (see, *e.g.*, Figure 1). To define the clones we introduce the following notions, where f is an n -ary Boolean function:

- f is *c-reproducing* if $f(c, \dots, c) = c$, $c \in \{0, 1\}$.
- f is *monotonic* if $a_1 \leq b_1, \dots, a_n \leq b_n$ implies $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$.

- f is *c-separating of degree k* if for all $A \subseteq f^{-1}(c)$ of size $|A| = k$ there exists an $i \in \{1, \dots, n\}$ such that $(a_1, \dots, a_n) \in A$ implies $a_i = c$, $c \in \{0, 1\}$.
- f is *c-separating* if f is *c-separating of degree $|f^{-1}(c)|$* .
- f is *self-dual* if $f \equiv \text{dual}(f)$, where $\text{dual}(f)(x_1, \dots, x_n) := \neg f(\neg x_1, \dots, \neg x_n)$.
- f is *affine* if $f \equiv x_1 \oplus \dots \oplus x_n \oplus c$ with $c \in \{0, 1\}$.

A list of some clones with definitions and finite bases is given in Table 1.

We will often add some function $f \notin C$ to a clone C and consider the clone $C' = [C \cup \{f\}]$ generated out of C and f . With Post's lattice one can determine this C' quite easily: It is the lowest clone above C that contains f . We will use in particular the identities $[S_{12} \cup \{C_1\}] = S_1$, $[D \cup \{C_1\}] = \text{BF}$, $[R_1 \cup \{C_0\}] = \text{BF}$, $[D_1 \cup \{C_1\}] = R_1$, and $[S_{10} \cup \{C_1\}] = M_1$.

A propositional formula using only connectives from B is called a *B-formula*.

Definition 1. Let f be an n -ary Boolean function and let B be a set of Boolean functions. A *B-formula* φ with $\text{Vars}(\varphi) = \{x_1, \dots, x_k\}$ is called *B-representation of f* if there is an index function $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$ such that $\forall x_1, \dots, x_k \in \{0, 1\}$ it holds $f(x_{\pi(1)}, \dots, x_{\pi(n)}) = 1$ if and only if φ evaluates to 1.

We note that such a *B-representation* exists for every $f \in [B]$. We note further that, if f does not contain fictive coordinates, then there is also a *B-representation* for f without fictive variables. We shall keep this in mind, since some problems we consider are not stable under introduction/elimination of fictive variables.

There is a canonical transformation of a B_1 -formula φ_1 into a *B-formula*, if $B_1 \subseteq [B]$: replace every connective in φ_1 by its *B-representation*. Though, this may lead to an explosion of the formula size. This can happen when a *B-representation* for some $f \in [B]$ uses some input variable more than once and φ_1 is of linear nesting depth, see e.g. [6]. We will nevertheless use this transformation idea in order to obtain reductions. This is possible since in the cases we encounter, we are always able to (re-)write φ_1 as a formula of logarithmic nesting depth. We note that this is not possible in general. We call formulæ of logarithmic nesting depth *compact*.

3 Enumeration without Order Prescription

We begin by looking at the model enumeration problem without order prescription.

Problem: ENUMSAT(B)

Instance: a *B-formula* φ

Question: generate all models of φ (without duplicates)

This problem has been studied in [2] considering *B-circuits* instead of *B-formulæ*. It is not difficult to observe that the algorithms from [2] also prove SpaceDelayP-membership for *B-formulæ* for the clones M , L , D and S_0^2 . But we have to slightly

Name	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R ₁	$\{f \mid f \text{ is 1-reproducing}\}$	$\{x \vee y, x = y\}$
R ₂	$R_0 \cap R_1$	$\{\vee, x \wedge (y = z)\}$
M	$\{f \mid f \text{ is monotonic}\}$	$\{x \vee y, x \wedge y, C_0, C_1\}$
S ₀ ⁿ	$\{f \mid f \text{ is 0-separating of degree } n\}$	$\{x \rightarrow y, t_2^{n+1}\}$
S ₀	$\{f \mid f \text{ is 0-separating}\}$	$\{x \rightarrow y\}$
S ₁	$\{f \mid f \text{ is 1-separating}\}$	$\{x \wedge \neg y\}$
S ₀₀ ⁿ	$S_0^n \cap R_2 \cap M$	$\{x \vee (y \wedge z), t_2^{n+1}\}$
S ₀₀	$S_0 \cap R_2 \cap M$	$\{x \vee (y \wedge z)\}$
S ₁₂	$S_1 \cap R_2$	$\{x \wedge (y \rightarrow z)\}$
S ₁₀	$S_1 \cap R_2 \cap M$	$\{x \wedge (y \vee z)\}$
D	$\{f \mid f \text{ is self-dual}\}$	$\{(x \wedge \neg y) \vee (x \wedge \neg z) \vee (\neg y \wedge \neg z)\}$
D ₁	$D \cap R_2$	$\{d_1\}$
D ₂	$D \cap M$	$\{t_2^3\}$
L	$\{f \mid f \text{ is affine}\}$	$\{x \oplus y, C_1\}$
V	$\{f \mid f \text{ is a disjunction of variables or constants}\}$	$\{x \vee y, C_0, C_1\}$
E	$\{f \mid f \text{ is a conjunction of variables or constants}\}$	$\{x \wedge y, C_0, C_1\}$

Table 1. List of relevant Boolean clones with definitions and bases, where t_p^q denotes the q -ary p -threshold function and $d_1(x, y, z) = (x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge \neg z)$.

modify the hardness proof from [2] in order to deal with the issue of possible exponential blowup. Hardness of ENUMSAT(B) is inherited from SAT^{*}(B), the non-trivial satisfiability problem for B -formulae (given a B -formula, does it admit a non-trivial model m , i.e., $m \notin \{0, 1\}$?). A look at Post's lattice shows us that $S_{12} \not\subseteq [B]$ if and only if either $[B] \subseteq M$, or $[B] \subseteq L$, or $[B] \subseteq D$, or $[B] \subseteq S_0^2$. The following proposition will therefore complete the classification.

Proposition 2. *Let $S_{12} \subseteq [B]$. Then SAT^{*}(B) is NP-complete.*

Theorem 3. *Let B be a finite set of Boolean functions. Then ENUMSAT(B) is*

1. NP-hard if $S_{12} \subseteq [B]$,
2. in SpaceDelayP otherwise (i.e., $[B] \subseteq M$ or $[B] \subseteq L$ or $[B] \subseteq D$ or $[B] \subseteq S_0^2$).

4 Enumeration by Order of Non-decreasing Weight

In this section we consider model enumeration by order of non-decreasing weight.

Problem: ENUMSAT_↑(B)

Instance: a B -formula φ

Question: generate all models of φ by order of non-decreasing weight

Proposition 4. *Let $[B] \subseteq V$ or $[B] \subseteq E$ or $[B] \subseteq L$ or $[B] \subseteq S_0$. Then ENUMSAT_↑(B) \in SpaceDelayP.*

Proof. The first three cases are easy. More interesting is the fourth case. Let φ be a B -formula with n variables. Since we are 0-separating, we know that there

hard according to [10]). The second and third case are technically involving, where we deal with the q -ary p -threshold function.

Theorem 6. *Let B be a finite set of Boolean functions. Then $\text{ENUMSAT}_\uparrow(B)$*

1. *is NP-hard if $S_{00}^n \subseteq [B]$ for some $n \geq 2$ or $D_2 \subseteq [B]$ or $S_{10} \subseteq [B]$,*
2. *is in SpaceDelayP otherwise (i.e., $[B] \subseteq S_0$ or $[B] \subseteq V$ or $[B] \subseteq L$ or $[B] \subseteq E$).*

We turn to the weighted variant.

The following method will deliver us several tractability results.

Theorem 7 (Priority queue method [9,16]). *Let $E = (I, \leq)$ be an enumeration problem. If it holds*

1. *for each $x \in I$, \leq restricted to $Sol(x)$ is total and computable in polynomial time in $|x|$,*
2. *it can be determined in polynomial time in $|x|$ whether $Sol(x)$ is non-empty and if so, then $\min(Sol(x))$ is computable in polynomial time in $|x|$,*
3. *there is a binary function f such that for all $x \in I$ and for all $y \in Sol(x)$ holds:*
 - (a) *$f(x, y)$ is computable in polynomial time in $|x|$*
 - (b) *$f(x, y) \subseteq Sol(x)$*
 - (c) *if $y \neq \min(Sol(x))$ then there is a $z \in Sol(x)$ such that $z < y$ and $y \in f(x, z)$,*

then $E \in \text{DelayP}$.

Proof. Correctness of the following algorithm is not difficult to observe.

```

1: if  $Sol(x) = \emptyset$  then return 'no'
2:  $Q = \text{newPriorityQueue}(\leq)$ 
3: compute  $\ell := \min(Sol(x))$ 
4:  $Q.\text{enqueue}(\ell)$ 
5: while  $Q$  is not empty do
6:    $\ell := Q.\text{dequeue}$ 
7:   output  $\ell$ 
8:   compute  $L := f(x, \ell)$ 
9:   for all  $z \in L$  do
10:    if  $z > \ell$  then  $Q.\text{enqueue}(z)$ 
11:   end for
12: end while

```

The priority queue is supposed to eliminate duplicates. Note that this method may run in exponential space.

In order to apply the method to the partial order induced by the weight of assignments, it suffices to extend it to a total order, for instance by the lexicographical order on assignments.

Proposition 8. *Let $[B] \subseteq V$ or $[B] \subseteq E$. Then $\text{W-ENUMSAT}_\uparrow(B) \in \text{DelayP}$.*

Proof. In the first case a B -formula can be seen as disjunction of variables and constants. All assignments are models, with the possible exception of $\mathbf{0}$. Thus, we reduce our problem to

Problem: SUBSETSUM

Instance: A sequence of non-negative integers $C = (w_1, \dots, w_n) \in \mathbb{N}^n$

Question: generate all subsets $S \subseteq \{1, \dots, n\}$ by non-decreasing weight $\delta(S)$, where $\delta(S) = \sum_{i \in S} w_i$

This task can be solved in polynomial delay and polynomial space by a dynamic programming method if the weights on the variables are polynomially bounded [5]. Otherwise, the priority queue method from Theorem 7 is applicable with $f(C, S) = \{S \cup \{i\} \mid i \in \{1, \dots, n\}\}$.

In the second case a B -formula can be seen as conjunction of variables and constants. If this disjunction contains a constant C_0 , then there are no models. Otherwise $\mathbf{1}$ is the only model, up to fictive variables occurring in constants C_1 . Again, we reduce our problem to SUBSETSUM as in the previous case.

Proposition 9. *Let $[B] \subseteq \mathbf{L}$. Then $\text{W-ENUMSAT}_{\uparrow}(B) \in \text{DelayP}$.*

Proof. Apply Theorem 7 with $f(\varphi, m) = \{m \cup \{x\} \mid x \text{ fictive}\} \cup \{m \cup \{x, y\} \mid x, y \text{ not fictive and } m \cap \{x, y\} = \emptyset\}$.

The following previously tractable fragment becomes intractable.

Proposition 10. *Let $S_{00} \subseteq [B]$. Then $\text{W-MIN-ONES}(B)$ is NP-hard.*

Proof. Via a reduction from $\text{MIN-ONES}(B \cup \{C_0\})$, replacing C_0 by a fresh variable of big weight.

Theorem 11. *Let B be a finite set of Boolean functions. Then $\text{W-ENUMSAT}_{\uparrow}(B)$*

1. *is NP-hard if $S_{00} \subseteq [B]$ or $D_2 \subseteq [B]$ or $S_{10} \subseteq [B]$,*
2. *is in DelayP otherwise (i.e., $[B] \subseteq \mathbf{V}$ or $[B] \subseteq \mathbf{L}$ or $[B] \subseteq \mathbf{E}$).*

5 Enumeration by Order of Non-increasing Weight

In this section we consider model enumeration by order of non-increasing weight.

Problem: ENUMSAT $_{\downarrow}(B)$

Instance: a B -formula φ

Question: generate all models of φ by order of non-increasing weight

Analogously to Proposition 4 we obtain SpaceDelayP-membership for disjunctive, conjunctive, affine, or 0-separating formulæ.

Proposition 12. *Let $[B] \subseteq \mathbf{V}$ or $[B] \subseteq \mathbf{E}$ or $[B] \subseteq \mathbf{L}$ or $[B] \subseteq S_0$. Then $\text{ENUMSAT}_{\downarrow}(B) \in \text{SpaceDelayP}$.*

For monotone formulæ in general we apply once more the priority queue method.

Proposition 13. *Let $[B] \subseteq \mathbf{M}$. Then $\text{ENUMSAT}_{\downarrow}(B) \in \text{DelayP}$.*

Proof. Apply Theorem 7 with $f(\varphi, m) = \{m \setminus \{x\} \mid m \setminus \{x\} \models \varphi\}$.

We now address one of the rare cases where the priority queue method is not applicable and still we obtain tractability. We use for this the following classical result from combinatorics.

Theorem 14 (Erdős-Ko-Rado Theorem [8]). *Let $n \geq 2r$ and A be a family of distinct subsets of $\{1, \dots, n\}$ such that each subset is of size r and each pair of subsets intersects. Then it holds*

$$|A| \leq \binom{n-1}{r-1}.$$

Lemma 15. *Let $f \in \mathbf{S}_0^2$ be an n -ary Boolean function and let k be an integer such that $n/2 \leq k \leq n$. Then the number of models of weight k is at least $\binom{n-1}{k-1}$.*

Proposition 16. *Let $[B] \subseteq \mathbf{S}_0^2$. Then $\text{ENUMSAT}_{\downarrow}(B) \in \text{DelayP}$.*

Proof. In a first step we give a description of the enumeration scheme for the weight range n down to $n/2$.

We start with weight n : there is one such assignment which is also a model (all functions in \mathbf{S}_0^2 are 1-reproducing). We continue with an inductive argument (for $n/2 \leq k < n$): assume that we know for weight k exactly the set of models S_k . By Lemma 15, we have $\binom{n-1}{k-1} \leq |S_k| \leq \binom{n}{k}$. The total time needed to output these models is something polynomial in $|S_k|$. This delivers enough time to brute-force all assignments of the next weight level $k-1$: There are $\binom{n}{k-1}$ such assignments to be tested, and the factor between $\binom{n}{k-1}$ and $|S_k|$ is obviously polynomially bounded in n . Summed up, while outputting (with polynomial delay) the models of weight k , we can compute the set of models of weight $k-1$. Repeated application of this allows to enumerate with polynomial delay all models in the weight range n down to $n/2$ by order of non-increasing weight.

The models in the weight range $n/2$ down to 0 can be computed and stored during the first step: When during the first step an assignment a is tested, also test its complement, \bar{a} , which lies then in the weight range $n/2$ down to 0. If \bar{a} is a model, put it on a stack. After step 1 has finished, output all the assignments from the stack.

We turn to the intractable cases. Solving $\text{ENUMSAT}_{\downarrow}$ requires to efficiently solve MAX-ONES^* , the task of computing a model of maximal weight different from $\mathbf{1}$. The hardness of this task will therefore deliver us hardness of $\text{ENUMSAT}_{\downarrow}$. The hardness of MAX-ONES^* is obtained from SAT^* and the following problem.

Problem: INVERSE-ROOT-WEIGHT-SAT

Instance: a 3CNF-formula φ of n variables
Question: does φ admit a model of weight $\geq n - \sqrt{n}$?

Lemma 17. INVERSE-ROOT-WEIGHT-SAT is NP-complete. It remains NP-complete if the number of variables is assumed to be a power of 3.

Proposition 18. Let $S_{12} \subseteq [B]$ or $D_1 \subseteq [B]$. Then $\text{MAX-ONES}^*(B)$ is NP-hard.

Proof. In the first case we reduce from $\text{SAT}^*(B)$ via $\varphi \mapsto (\varphi, 1)$ and conclude with Proposition 2. In the second case we have a technically involving reduction from INVERSE-ROOT-WEIGHT-SAT.

Theorem 19. Let B be a finite set of Boolean functions. Then $\text{ENUMSAT}_{\downarrow}(B)$

1. is NP-hard if $S_{12} \subseteq [B]$ or $D_1 \subseteq [B]$,
2. is in DelayP otherwise (i.e., $[B] \subseteq S_0^2$ or $[B] \subseteq M$ or $[B] \subseteq L$), where $\text{ENUMSAT}_{\downarrow}(X) \in \text{SpaceDelayP}$ for $X \in \{V, E, L, S_0\}$

Lastly, a look at the weighted variant, where we obtain only partial results.

Proposition 20. Let $[B] \subseteq S_0$ or $[B] \subseteq M$. Then $\text{W-ENUMSAT}_{\downarrow}(B) \in \text{DelayP}$.

Proof. Apply Theorem 7 with $f(\varphi, m) = \{m \setminus \{x\} \mid m \setminus \{x\} \models \varphi\}$.

Proposition 21. Let $[B] \subseteq L$. Then $\text{W-ENUMSAT}_{\downarrow}(B) \in \text{DelayP}$.

Proof. Analogously to Proposition 9.

The following tractability indicates that also $\text{W-ENUMSAT}_{\downarrow}(S_0^2)$ might be tractable. However, none of the above algorithmic strategies seems to work out.

Proposition 22. Let $[B] \subseteq S_0^2$. Then $\text{W-MAX-ONES}^*(B) \in P$.

6 Conclusion

In this paper we provided complete complexity classifications of the problem of enumerating all satisfying assignments of a propositional B -formula for every set B of allowed connectives, imposing the orders of non-decreasing weight and non-increasing weight. We also considered the weighted variant, where the variables are assigned a non-negative integer weight. We obtained a complete classification for the weighted variant when imposing the order of non-decreasing weight and remained with one open case for the order of non-increasing weight when the connectives are 0-separating of degree 2. Interesting are the polynomial delay algorithms we obtained. They either rely on combinatorial bounds allowing a brute force approach, or on the use of a priority queue which necessarily leads to an exponential space usage. Future research could affront the open case, but should also investigate the question of exponential space: can it be avoided, or is it inherent to these problems, in particular to SUBSETSUM without polynomial bounds on the weights?

Acknowledgements. The author would like to thank Johan Thapper for combinatorial support.

References

1. O. Beyersdorff, A. Meier, M. Thomas, and H. Vollmer. The complexity of propositional implication. *Inf. Process. Lett.*, 109(18):1071–1077, 2009.
2. E. Böhrer, N. Creignou, M. Galota, S. Reith, H. Schnoor, and H. Vollmer. Complexity classifications for different equivalence and audit problems for boolean circuits. *Logical Methods in Computer Science*, 8(3), 2012.
3. A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, M. Jerrum, and D. Richerby. The complexity of weighted and unweighted $\#csp$. *J. Comput. Syst. Sci.*, 78(2):681–688, 2012.
4. N. Creignou and J.-J. Hébrard. On Generating All Solutions of Generalized Satisfiability Problems. *ITA*, 31(6):499–511, 1997.
5. N. Creignou, F. Olive, and J. Schmidt. Enumerating all solutions of a Boolean CSP by non-decreasing weight. In *Proc. of 14th International Conference on Theory and Applications of Satisfiability Testing (SAT'2011)*, Lecture notes in computer science, pages 120–133, 2011.
6. N. Creignou, J. Schmidt, and M. Thomas. Complexity classifications for propositional abduction in post’s framework. *J. Log. Comput.*, 22(5):1145–1170, 2012.
7. A. Durand, M. Hermann, and P. G. Kolaitis. Subtractive reductions and complete problems for counting complexity classes. *Theor. Comput. Sci.*, 340(3):496–513, 2005.
8. P. Erdős, Chao Ko, and R. Rado. Intersection theorem for system of finite sets. *Quart. J. Math. Oxford Ser.*, 12:313–318, 1961.
9. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.
10. S. Khanna, M. Sudan, and D. P. Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. In *STOC*, pages 11–20, 1997.
11. B. Kimelfeld and Y. Sagiv. Incrementally computing ordered answers of acyclic conjunctive queries. In *NGITS*, pages 141–152, 2006.
12. H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.
13. E. Post. The two-valued iterative systems of mathematical logic. *Ann. Math. Stud.*, 5:1–122, 1941.
14. S. Reith. On the complexity of some equivalence problems for propositional calculi. In *Proc. 28th MFCS*, volume 2747 of *LNCS*, pages 632–641, 2003.
15. T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.
16. J. Schmidt. Enumeration: Algorithms and complexity. Preprint (2009), available at <http://www.thi.uni-hannover.de/fileadmin/forschung/arbeiten/schmidt-da.pdf>.
17. H. Schnoor and I. Schnoor. Enumerating all solutions for constraint satisfaction problems. In *STACS*, pages 694–705, 2007.
18. C. Schnorr. Optimal algorithms for self-reducible problems. In *ICALP*, pages 322–337, 1976.
19. Y. Strobecki. Enumeration complexity and matroid decomposition. *Phd thesis*, 2010.
20. M. Thomas. The complexity of circumscriptive inference in Post’s lattice. In *Proc. 10th LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, pages 290–302. Springer Verlag, 2009.

7 Appendix

Proof of Proposition 2
 Proof of Proposition 4
 Proof of Proposition 5
 Proof of Proposition 10
 Proof of Lemma 15
 Proof of Lemma 17
 Proof of Proposition 18
 Proof of Proposition 22

Proof of Proposition 2 ($\text{SAT}^*(B)$ is NP-complete if $S_{12} \subseteq [B]$)

Proof. NP-membership is obvious. For the hardness, we give a reduction from the satisfiability problem for B -formulae. We know from [12] that $\text{SAT}(B')$ is NP-complete if $S_1 \subseteq [B']$. Since $S_1 = [S_{12} \cup \{C_0\}] \subseteq [B \cup \{C_0\}]$, we conclude that $\text{SAT}(B \cup \{C_0\})$ is NP-complete. Let φ be a $B \cup \{C_0\}$ -formula with variable set x_1, \dots, x_n . We construct $\varphi' = \varphi[C_0/f] \wedge \bigwedge_{i=1}^n t \wedge (f \rightarrow x_i)$. It is not difficult to verify that φ is satisfiable if and only if φ' admits a non-trivial model. Note that \wedge and $x \wedge (y \rightarrow z)$ are in S_{12} and have therefore a B -representation. We transform φ' into the final B -formula by replacing the connectives \wedge and $x \wedge (y \rightarrow z)$ with their B -representations. We avoid exponential blowup by a compact φ' : write the n -ary conjunction as a balanced tree of the binary conjunction \wedge .

Proof of Proposition 4 ($\text{ENUMSAT}_\uparrow(X) \in \text{SpaceDelayP}$ for $X \in \{V, E, L, S_0\}$)

Proof. In the first case a B -formula can be seen as disjunction of variables and constants. All assignments are models, with the possible exception of $\mathbf{0}$. We obviously can enumerate those assignments by order of non-decreasing weight by standard combinatorial methods.

In the second case a B -formula can be seen as conjunction of variables and constants. If this disjunction contains a constant C_0 , then there are no models. Otherwise $\mathbf{1}$ is the only model, up to fictive variables occurring in constants C_1 . Again we can enumerate those assignments by order of non-decreasing weight by standard combinatorial methods.

In the third case a B -formula can be seen as linear equation over $\text{GF}(2)$. Therefore, the set of models is either the set of assignments with an even number of non-fictive variables set to 1, or the set with an odd number of non-fictive variables set to 1. Again, all these models can be enumerated by non-decreasing weight by standard combinatorial methods.

The fourth case is treated in the paper.

Proof of Lemma 15 (Let $f \in S_0^2$ be an n -ary Boolean function and let k be an integer such that $n/2 \leq k \leq n$. Then the number of models of weight k is at least $\binom{n-1}{k-1}$.)

Proof. Since the functions of S_0^2 are 0-separating of degree 2, the statement of this lemma is nothing else than a disguised form of the Erdős-Ko-Rado Theorem [8]: Set $r = n - k$ and represent an assignment as subset of $\{1, \dots, n\}$ by the set of indexes of the coordinates which are set to 0. Then A corresponds to the set of non-models.

Preparations for Proof of Proposition 5 (MIN-ONES(B) is NP-hard if $S_{10} \subseteq [B]$ or $S_{00}^n \subseteq [B]$ for an $n \geq 2$ or $D_2 \subseteq [B]$)

Lemma 23. *Let $E_2 \subseteq [B]$. Then $\text{MIN-ONES}(B \cup \{C_1\}) \leq_m^{\log} \text{MIN-ONES}(B)$.*

Proof. We map φ to $\varphi' = \varphi[C_1/t] \wedge t$ and k to $k' = k + 1$. The formula φ' can be written as a B -formula by replacing the connective \wedge by its B -representation ($\wedge \in E_2 \subseteq [B]$).

Definition 24. *We denote by t_q^p the p -ary q -threshold function, $p > q \geq 2$. Define ${}^d\psi_q^p$ to be a complete t_q^p -tree of depth d .*

Lemma 25. *The formula ${}^d\psi_q^p$*

1. *has arity p^d ,*
2. *evaluates to 0 whenever less than q^d inputs are set to 1, and*
3. *evaluates to 1 whenever more than $p^d - q^d$ inputs are set to 1.*

Proof. Obviously ${}^d\psi_q^p$ has arity p^d .

We prove the second statement by induction over d . For $d = 1$ we have ${}^1\psi_q^p = t_q^p$ and the statement holds by definition of t_q^p . Consider then ${}^{(d+1)}\psi_q^p = t_q^p({}^d\psi_q^p, \dots, {}^d\psi_q^p)$. Assuming that none of the ${}^d\psi_q^p$ can be triggered with less than q^d inputs set to 1, we observe that ${}^{(d+1)}\psi_q^p$ cannot be triggered with less than $q \cdot q^d = q^{d+1}$ inputs set to 1.

The third statement follows from the second by the self-duality of ${}^d\psi_q^p$.

Lemma 26. *If $t_q^p \in [B]$ for some $p, q \in \mathbb{N}$ with $p > q \geq 2$, then $\text{MIN-ONES}(B \cup \{C_0\}) \leq_m^{\log} \text{MIN-ONES}(B)$.*

Proof. Let (φ, k) be an instance of $\text{MIN-ONES}(B \cup \{C_0\})$, n the number of variables in φ . Choose $d \in \mathbb{N}$ such that $q^{d-1} < n < q^d$ (i.e., $d - 1 < \log_q(n) < d$) and we obtain with Lemma 25 that ${}^d\psi_q^p$ has arity and size polynomial in n and evaluates to 0 whenever less than $n + 1$ inputs are set to 1 ($n + 1 \leq q^d$).

Denote by ${}^dF_q^p$ the B -formula obtained from ${}^d\psi_q^p$ by replacing each t_q^p by its B -representation. Exponential blowup does not occur since ${}^d\psi_q^p$ is compact. We finally map (φ, k) to (φ', k') , where $\varphi' = \varphi[C_0/{}^dF_q^p(y_1, \dots, y_{p^d})]$ and $k' = \min(n, k)$ and the y_1, \dots, y_{p^d} are fresh variables. One easily verifies that φ admits a model of weight $\leq k$ if and only if φ' admits a model of weight $\leq \min(n, k)$.

Proof of Proposition 5 (MIN-ONES(B) is NP-hard if $S_{10} \subseteq [B]$ or $S_{00}^n \subseteq [B]$ for an $n \geq 2$ or $D_2 \subseteq [B]$)

Proof. In the first case we have $E_2 \subseteq S_{10} \subseteq [B]$, so we obtain by Lemma 23 that $\text{MIN-ONES}(B \cup \{C_1\}) \leq_m^{\log} \text{MIN-ONES}(B)$. Since $\{\wedge, \vee\} \subseteq M_1 = [S_{10} \cup \{C_1\}] \subseteq [B \cup \{C_1\}]$, we can reduce from $\text{MIN-ONES}(\text{positive-2CNF})$ (NP-hard according to [10]) to $\text{MIN-ONES}(B \cup \{C_1\})$ by replacing every connective (\wedge, \vee) by its $(B \cup \{C_1\})$ -representation, avoiding exponential blowup by a compact representation of the multi-ary conjunction in the 2CNF-formula.

In the second case we have that $t_2^{n+1} \in [B]$, in the third case we have that $t_2^3 \in [B]$. In both cases we obtain by Lemma 26 that $\text{MIN-ONES}(B \cup \{C_0\}) \leq_m^{\log} \text{MIN-ONES}(B)$. Since further in both cases it holds $\{\wedge, \vee\} \subseteq M_2 \subseteq [B \cup \{C_0\}]$, we can reduce from $\text{MIN-ONES}(\text{positive-2CNF})$ to $\text{MIN-ONES}(B \cup \{C_0\})$ by replacing every connective (\wedge, \vee) by its $(B \cup \{C_0\})$ -representation (avoiding exponential blowup by a compact formula).

Proof of Proposition 10 ($\text{W-MIN-ONES}(B)$ is NP-hard)

Proof. We give a reduction from $\text{MIN-ONES}(B \cup \{C_0\})$, an NP-hard problem according to Theorem 6, since $D_2 \subseteq M_0 = [S_{00} \cup \{C_0\}] \subseteq [B \cup \{C_0\}]$. Let (φ, k) be an instance of $\text{MIN-ONES}(B \cup \{C_0\})$, n the number of variables in φ . We map (φ, k) to (φ', k') , where $\varphi' = \varphi[C_0/f]$, f a fresh variable and $k' = \min(n, k)$. We set the weight of all variables from φ to 1 and the weight of f to $n + 1$. One easily verifies that φ admits a model of weight $\leq k$ if and only if φ' admits a model of weight $\leq \min(n, k)$.

Proof of Lemma 17 ($\text{INVERSE-ROOT-WEIGHT-SAT}$ is NP-complete)

Proof. The NP-membership is obvious: guess an assignment of weight $\geq n - \sqrt{n}$ and verify whether it is a satisfying one.

By flipping all literals in an instance of $\text{INVERSE-ROOT-WEIGHT-SAT}$, we obtain equivalence to

Problem: ROOT-WEIGHT-SAT
Instance: a 3CNF-formula φ of n variables
Question: does φ admit a model of weight $\leq \sqrt{n}$?

We give a reduction from $\text{MIN-ONES}(3\text{CNF})$ (NP-hard according to [10]) to ROOT-WEIGHT-SAT . Let (φ, k) be an instance of $\text{MIN-ONES}(3\text{CNF})$, let n be the number of variables of φ .

- If we have $k \geq \sqrt{n}$, we set $\ell = \min(k, n)$ and $r = \ell^2 - n$ and map the instance (φ, k) to $\varphi' = \varphi \wedge \neg y_1 \wedge \dots \wedge \neg y_r$.

Let $(\varphi, k) \in \text{MIN-ONES}$. By definition of ℓ this implies that $(\varphi, \ell) \in \text{MIN-ONES}$. That is, there is a model σ of weight $w(\sigma) \leq \ell$. φ' has $n' = n + r = \ell^2$ variables. The model σ can be extended to a model σ' of φ' by setting additionally the variables y_1, \dots, y_r to 0. This σ' has then weight

$$w(\sigma') = w(\sigma) \leq \ell = \sqrt{n'}.$$

That is, $\varphi' \in \text{ROOT-WEIGHT-SAT}$.

Conversely, let $\varphi' \in \text{ROOT-WEIGHT-SAT}$. That is, there is a model σ' with

$$w(\sigma') \leq \sqrt{n'} = \ell$$

Construct σ from σ' by removing the assignments for the y_i and we get that σ satisfies φ and $w(\sigma) \leq \ell$. That is, $(\varphi, \ell) \in \text{MIN-ONES}$ and, since $\ell \leq k$, $(\varphi, k) \in \text{MIN-ONES}$.

– Else if $k < \sqrt{n}$, we set $t = \sqrt{n - k + \frac{1}{4}} - k + \frac{1}{2}$. One easily verifies that

$$k + t = \sqrt{n + t} \quad (1)$$

and that consequently

$$\sqrt{n + \lfloor t \rfloor} \leq \sqrt{n + t} = k + t < k + \lfloor t \rfloor + 1 \quad (2)$$

From the property that for $x \geq 0$ we have $\lfloor \sqrt{x} \rfloor \leq \sqrt{\lfloor x \rfloor}$ one obtains that

$$k + \lfloor t \rfloor = \lfloor k + t \rfloor = \lfloor \sqrt{n + t} \rfloor \leq \sqrt{\lfloor n + t \rfloor} = \sqrt{n + \lfloor t \rfloor} \quad (3)$$

Set now $r = \lfloor t \rfloor$ and map the instance (φ, k) to $\varphi' = \varphi \wedge y_1 \wedge \dots \wedge y_r$.

Let $(\varphi, k) \in \text{MIN-ONES}$. That is, there is a model σ of weight $w(\sigma) \leq k$. φ' has $n' = n + r$ variables. The model σ can be extended to a model σ' of φ' by setting additionally the variables y_1, \dots, y_r to 1. This σ' has then weight

$$w(\sigma') = w(\sigma) + r \leq k + r \stackrel{(3)}{\leq} \sqrt{n + r} = \sqrt{n'}.$$

That is, $\varphi' \in \text{ROOT-WEIGHT-SAT}$.

Conversely, let $\varphi' \in \text{ROOT-WEIGHT-SAT}$. That is, there is a model σ' with $w(\sigma') \leq \sqrt{n'} = \sqrt{n + r} \stackrel{(2)}{<} k + r + 1$. That is, $w(\sigma') \leq k + r$. By construction of φ' , we know that σ' has to set all y_1, \dots, y_r to 1. Thus, by reducing σ' by the values for the y_i , we obtain an assignment σ with $w(\sigma) \leq k$. By construction of φ' , we conclude that σ is indeed a model of φ . That is, $(\varphi, k) \in \text{MIN-ONES}$.

To prove the second statement, add enough dummy variables to fill up to the next power of 3, be d such that $3^{d-1} < n \leq 3^d$. Force the right amount of them to 1 and the rest to 0 in order to map models of weight $\leq \sqrt{n}$ to models of weight $\leq \sqrt{3^d}$ and models of weight $> \sqrt{n}$ to models of weight $> \sqrt{3^d}$.

Proof of Proposition 18 ($\text{MAX-ONES}^*(B)$ is NP-hard if $S_{12} \subseteq [B]$ or $D_1 \subseteq [B]$)

Proof. In the first case we reduce from $\text{SAT}^*(B)$ via $\varphi \mapsto (\varphi, 1)$ and conclude with Proposition 2.

In the second case we give a reduction from INVERSE-ROOT-WEIGHT-SAT with the assumption that the number of variables is a power of 3. Let φ be an instance of INVERSE-ROOT-WEIGHT-SAT with variable set x_1, \dots, x_n , where $n = 3^d$. We transform φ in several steps. We assume φ to be a compact $\{\wedge, \vee, \neg\}$ -formula.

1. Since $\{\wedge, \vee, \neg\} \subseteq \text{BF} = [\mathbf{R}_1 \cup \{C_0\}]$ and $\mathbf{R}_1 = [\vee, =]$, we can transform φ into a compact $\{\vee, =, C_0\}$ -formula φ_1 by replacing the connectives $\{\wedge, \vee, \neg\}$ with their $\{\vee, =, C_0\}$ -representations.
2. Set $\varphi_2 = \varphi_1[C_0/f] \wedge \bigwedge_{i=1}^n (f \rightarrow x_i)$, where f is a fresh variable and the n -ary conjunction be represented in a compact way by the binary conjunction. Note that φ_2 is now a compact $\{\vee, =, \wedge, \rightarrow\}$ -formula.
3. Since $\{\vee, =, \wedge, \rightarrow\} \subseteq \mathbf{R}_1 = [\mathbf{D}_1 \cup \{C_1\}]$ and $\mathbf{D}_1 = [d_1]$, we can transform φ_2 into a compact $\{d_1, C_1\}$ -formula φ_3 by replacing the connectives $\{\vee, =, \wedge, \rightarrow\}$ with their $\{d_1, C_1\}$ -representations.
4. Set $\varphi_4 = \varphi_3[C_1/d_1^3 \psi_2^3(x_1, \dots, x_{3^d})]$. Observe that φ_4 is a compact $\{d_1, t_2^3\}$ -formula of size polynomial in the size of φ with variable set x_1, \dots, x_n, f .
5. At last, since $d_1, t_2^3 \in \mathbf{D}_1 \subseteq [B]$, we can transform φ_4 into φ_5 by replacing the connectives d_1, t_2^3 with their B -representations. Note that φ_5 is still of polynomial size, since φ_4 is compact.

We finally map φ to $(\varphi_5, \lceil n - \sqrt{n} \rceil)$. It is not difficult to verify that φ admits a model of weight w with $\lceil n - \sqrt{n} \rceil \leq w \leq n$ if and only if φ_5 admits a model of weight v with $\lceil n - \sqrt{n} \rceil \leq v < n + 1 = |\text{Vars}(\varphi_5)|$ (to pass from one model to another, add/remove the assignment $f = 0$).

Proof of Proposition 22 ($\text{W-MAX-ONES}^*(\mathbf{S}_0^2) \in \text{P}$)

Proof. Let n be the number of variables in an instance. We obtain a model different from $\mathbf{1}$ of maximal weight by searching among the set of assignments with one 0 and $(n - 1)$ 1's. The number of such assignments is obviously polynomial in n and the property of being 0-separating of degree 2 guarantees us among them a non-empty set of models. Obviously a maximum weight model different from $\mathbf{1}$ is among them (all assignments with more than one 0 are of less or equal weight).